

[illegible]


```
1 0001 0 MODULE setdir (
2 0002 0 IDENT = 'V04-000',
3 0003 0 ADDRESSING_MODE(EXTERNAL=GENERAL,
4 0004 0 NONEXTERNAL=LONG_RELATIVE)
5 0005 0 ) =
6 0006 1 BEGIN
7 0007 1
8 0008 1
9 0009 1 *****
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 * ALL RIGHTS RESERVED.
14 0014 1 *
15 0015 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
16 0016 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
17 0017 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
18 0018 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
19 0019 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
20 0020 1 * TRANSFERRED.
21 0021 1 *
22 0022 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
23 0023 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
24 0024 1 * CORPORATION.
25 0025 1 *
26 0026 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
27 0027 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
28 0028 1 *
29 0029 1 *****
30 0030 1
31 0031 1
32 0032 1
33 0033 1 ++
34 0034 1 FACILITY: Set Directory Command
35 0035 1
36 0036 1 ABSTRACT:
37 0037 1
38 0038 1 This module processes the Set Directory command.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 Vax native, privileged user mode
43 0043 1
44 0044 1 --
45 0045 1
46 0046 1 AUTHOR: Gerry Smith CREATION DATE: 3-Nov-1981
47 0047 1
48 0048 1 MODIFIED BY:
49 0049 1
50 0050 1 V03-007 AEW0001 Anne E. Warner 6-Mar-1984
51 0051 1 Add error messages to prevent the use of search lists
52 0052 1 with devices for directories.
53 0053 1 FILSPCSRCH - File specification contains a searchlist
54 0054 1 NOSRCHLST - Command does not support search lists
55 0055 1 Both are defined in SHRMSG.MSG.
56 0056 1
57 0057 1 V03-006 GAS0170 Gerry Smith 23-Aug-1983
```


SETDIR
V04-000

I 16
16-Sep-1984 00:59:34
14-Sep-1984 12:09:06

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETDIR.B32;1

Page 2
(1)

58 0058 1
59 0059 1
60 0060 1
61 0061 1
62 0062 1
63 0063 1
64 0064 1
65 0065 1
66 0066 1
67 0067 1
68 0068 1
69 0069 1
70 0070 1
71 0071 1
72 0072 1
73 0073 1
74 0074 1
75 0075 1
76 0076 1
77 0077 1
78 0078 1
79 0079 1
80 0080 1
81 0081 1 **

Instead of simply doing an IOS_ACCESS to get the
directory's header, perform an actual open, by
issuing an IOS_ACCESS OR IOSM_ACCESS, to insure
proper file locking by the XQP.

V03-005 GAS0116 Gerry Smith 6-Apr-1983
Use the common qualifier routines.

V03-004 GAS0112 Gerry Smith 29-Mar-1983
Fix for the new CLI interface, as well as the new
command dispatcher.

V03-003 GAS0091 Gerry Smith 19-Oct-1982
Change input request for new CLD syntax.

V03-002 GAS0026 Gerry Smith 18-Dec-1981
Use shared message file, and lower fatal messages to
simple error messages.

V03-001 GAS0024 Gerry Smith 14-Dec-1981
Fix /LOG logic

SETDIR
V04-000

J 16
16-Sep-1984 00:59:34
14-Sep-1984 12:09:06

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETDIR.B32;1

Page 3
(2)

:	83	0082	1	LIBRARY 'SYSS\$LIBRARY:LIB';
:	84	0083	1	LIBRARY 'SYSS\$LIBRARY:TPAMAC';
:	85	0084	1	

```

87 0085 1 FORWARD ROUTINE
88 0086 1   set$directory : NOVALUE,      ! Main routine for directory
89 0087 1   get_qual,                  ! Get qualifiers and values
90 0088 1   set_attributes,            ! Routine to set directory attributes
91 0089 1   get_dir;                   ! Routine to get each directory
92 0090 1
93 0091 1
94 0092 1 EXTERNAL ROUTINE
95 0093 1   cli$present,                ! Determine presence of qualifier
96 0094 1   cli$get_value,              ! Get value of qualifier
97 0095 1   lib$qual_file_parse,        ! Get common qualifiers
98 0096 1   lib$qual_file_match,        ! See if directory matches common quals
99 0097 1   lib$file_scan,              ! Routine to get next directory
100 0098 1   check_privilege : NOVALUE, ! Routine to check for privilege
101 0099 1   search_error,              ! Where to go if file search fails
102 0100 1   file_error,                ! Where to go if file error occurs
103 0101 1   parse_uic,                 ! Routine for parsing UIC strings
104 0102 1   lib$tparse,                 ! Parser
105 0103 1   lib$cvd_dtb,                ! Convert decimal to number
106 0104 1   sys$fao;                   ! Formatted ASCII output
107 0105 1
108 0106 1
109 0107 1   ! Declare external data
110 0108 1
111 0109 1 EXTERNAL
112 0110 1   setfile$flags : BITVECTOR[32], ! Qualifier flags word
113 0111 1   uic_value,                  ! Owner UIC
114 0112 1   group,                      ! UIC group number
115 0113 1   member,                     ! UIC member number
116 0114 1   vrsn_value,                 ! Version limit
117 0115 1
118 0116 1   ! RMS structures
119 0117 1
120 0118 1   file_result : VECTOR[nam$c_maxrss,BYTE], ! Resultant name string
121 0119 1   file_expanded : VECTOR[nam$c_maxrss,BYTE], ! Expanded name string
122 0120 1   ! file rlf is in own section because it not externally defined.
123 0121 1   file_nam : $BBLOCK[nam$c_bln], ! File name block
124 0122 1   file_fab : $BBLOCK[fab$c_bln]; ! File FAB
125 0123 1
126 0124 1
127 0125 1   ! Error messages
128 0126 1
129 0127 1 EXTERNAL LITERAL
130 0128 1   lib$_filfaimat,              ! No match for common quals
131 0129 1   lib$_quipro,                 ! Quit processing
132 0130 1   set$_facility,                ! Facility code
133 0131 1   set$_closeerr,                ! Could not close file
134 0132 1   set$_entered,                ! File entered in a directory
135 0133 1   set$_enterr,                  ! Error entering file
136 0134 1   set$_modified,               ! File/directory modified
137 0135 1   set$_notdir,                  ! Not a directory
138 0136 1   set$_notods2,                 ! Not an ODS2 structure
139 0137 1   set$_readerr,                 ! Could not read file
140 0138 1   set$_remerr,                  ! Could not remove file
141 0139 1   set$_removed,                 ! Directory entry removed
142 0140 1   set$_writeerr;                ! Could not write to file
143 0141 1
```

```
144 0142 1 |
145 0143 1 | Declare some shared messages
146 0144 1 |
147 P 0145 1 | $SHR_MSGDEF (SET,119,LOCAL,
148 P 0146 1 | (syntax, error),
149 P 0147 1 | (openin, error),
150 P 0148 1 | (valerr, error),
151 P 0149 1 | (filspcsrc, error),
152 P 0150 1 | (nosrchlst, error));
153 0151 1 |
154 0152 1 |
155 0153 1 |
156 0154 1 | Literal data definitions
157 0155 1 |
158 0156 1 | LITERAL
159 0157 1 | true = 1;
160 0158 1 | false = 0;
161 0159 1 |
162 P 0160 1 | LITERAL
163 P 0161 1 | $EQLST
164 P 0162 1 | (QUAL-,1,1,
165 P 0163 1 | (quit,);
166 P 0164 1 | (log,);
167 P 0165 1 | (owner,);
168 P 0166 1 | (vrsn,));
169 0167 1 |
170 0168 1 |
171 0169 1 | Declare the context longword
172 0170 1 |
173 0171 1 | OWN
174 0172 1 | file_rlf : $NAM(),
175 0173 1 | context;
176 0174 1 |
```

```
| quit processing
| LOG bit
| OWNER UIC bit
| VERSION_LIMIT bit
```


SETDIR
V04-000

M 16
16-Sep-1984 00:59:34
14-Sep-1984 12:09:06

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETDIR.B32;1

Page 6
(4)

```

: 178      0175 1 |
: 179      0176 1 | TPARSE table for directories
: 180      0177 1 |
: 181      0178 1 $INIT_STATE (dir_state,dir_keys);
: 182      0179 1 |
: 183      P 0180 1 $STATE (dirstart,
: 184      0181 1 (tpa$_octal,...,group));      ! Get the group number
: 185      0182 1 |
: 186      P 0183 1 $STATE (
: 187      0184 1 (f,'));      ! Get rid of the comma
: 188      0185 1 |
: 189      P 0186 1 $STATE (
: 190      0187 1 (tpa$_octal,tpa$_exit,,,member));      ! Get the member number
: 191      0188 1 |
: 192      0189 1 PSECT OWN = $OWNS;
: 193      0190 1 PSECT GLOBAL = $GLOBALS;
```



```
195 0191 1 GLOBAL ROUTINE set$directory : NOVALUE =
196 0192 1 **
197 0193 1
198 0194 1 Functional description
199 0195 1
200 0196 1 This is the main control module for SET DIRECTORY. It calls
201 0197 1 LIB$FILE_SCAN to perform the necessary functions on the directory
202 0198 1 specified in the call to SET.
203 0199 1
204 0200 1 Calling sequence
205 0201 1
206 0202 1 CALL set$directory()
207 0203 1
208 0204 1 Input parameters
209 0205 1 none
210 0206 1
211 0207 1 Output parameters
212 0208 1 none
213 0209 1
214 0210 1 Implicit outputs
215 0211 1 none
216 0212 1
217 0213 1 Routine value
218 0214 1 none
219 0215 1
220 0216 1 Side effects
221 0217 1 none
222 0218 1
223 0219 1 --
224 0220 2 BEGIN
225 0221 2
226 0222 2
227 0223 2 LOCAL
228 0224 2 status,
229 0225 2 dir_desc : $BBLOCK[dsc$c_s_bln];
230 0226 2
231 0227 2
232 0228 2 Check that the image is running with appropriate privilege.
233 0229 2
234 0230 2 check_privilege();
235 0231 2
236 0232 2
237 0233 2 Get the common qualifiers.
238 0234 2
239 0235 2 status = lib$qual_file_parse(%REF(lib$m_cqf_exclude OR
240 0236 2 lib$m_cqf_before OR
241 0237 2 lib$m_cqf_since OR
242 0238 2 lib$m_cqf_created OR
243 0239 2 lib$m_cqf_modified OR
244 0240 2 lib$m_cqf_expired OR
245 0241 2 lib$m_cqf_backup OR
246 0242 2 lib$m_cqf_confirm OR
247 0243 2 lib$m_cqf_byowner),
248 0244 2 context);
249 0245 2 IF NOT .status
250 0246 2 THEN (SIGNAL(.status); RETURN);
251 0247 2
```

```

.TITLE SETDIR
.IDENT \V04-000\

.PSECT _LIB$STATES,NOWRT, SHR, PIC,1

TE::
.BLK 0
T:
.BLK 0
PE
.WORD 17908
DR
.LONG <<GROUP-U.3>-4>
PE
.WORD 1068
PE
.WORD 22004
DR
.LONG <<MEMBER-U.6>-4>
RGET
.WORD -1

.PSECT _LIB$KEYOS,NOWRT, SHR, PIC,1

S::
.BLK 0
Y0
.BLK 0

.PSECT $OWNS,NOEXE,2

```

```

02 00000 FILE_RLF:
    60 00001 .BYTE 2
    00 00002 .BYTE 96
    00 00003 .BYTE 0
000000 00 00004 .LONG 0
    00 00008 .BYTE 0
    00 00009 .BYTE 0
    00 0000A .BYTE 0
    00 0000B .BYTE 0
00000000 0000C .LONG 0
00000000 00010 .LONG 0
    0000# 00014 .WORD 0[B]
    0000# 00024 .WORD 0[3]
    0000# 0002A .WORD 0[3]
00000000 00030 .LONG 0
00000000 00034 .LONG 0
    00 00038 .BYTE 0
    00 00039 .BYTE 0
    00 0003A .BYTE 0
    00 0003B .BYTE 0
    00 0003C .BYTE 0
    00 0003D .BYTE 0
    00# 0003E .BYTE 0[2]
00000000 00040 .LONG 0
00000000 00044 .LONG 0
00000000 00048 .LONG 0
00000000 0004C .LONG 0
00000000 00050 .LONG 0
00000000 00054 .LONG 0
00000000# 00058 .LONG 0[2]
00060 CONTEXT: .BLKB 4

```

```

.EXTRN CLISPRESNT, CLISGET_VALUE
.EXTRN LIB$QUAL_FILE_PARSE
.EXTRN LIB$QUAL_FILE_MATCH
.EXTRN LIB$FILE_SCAN, CHECK_PRIVILEGE
.EXTRN SEARCH_ERROR, FILE_ERROR
.EXTRN PARSE_OIC, LIB$TPARSE
.EXTRN LIB$CVT_DTB, SYSS$FAO
.EXTRN SETFILE$FLAGS, UIC_VALUE
.EXTRN GROUP_MEMBER, VR$N_VALUE
.EXTRN FILE_RESULT, FILE_EXPANDED
.EXTRN FILE_NAM, FILE_FAB
.EXTRN LIB$FILFAMAT, LIB$QUIPRO
.EXTRN SET$FACILITY, SET$CLOSEERR
.EXTRN SET$ENTERED, SET$ENTERR
.EXTRN SET$MODIFIED, SET$NOTDIR
.EXTRN SET$NOTODS2, SET$READERR
.EXTRN SET$REMERR, SET$REMOVED
.EXTRN SET$WRITEERR

```

.PSECT \$CODE\$,NOWRT,2

```

.ENTRY SET$DIRECTORY, Save nothing
SUBL2 #12, SP
CALLS #0, CHECK_PRIVILEGE

```

00000000G SE
00

0000 00000
0C C2 00002
00 FB 00005

: 0191
: 0230

SETDIR
V04-000

E 1
16-Sep-1984 00:59:34 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:09:06 [CLIUTL.SRC]SETDIR.B32;1

Page 10
(5)

04	AE	00000000'	EF	9F	0000C	PUSHAB	CONTEXT	0235
		01FF	8F	3C	0C012	MOVZWL	#511, 4(SP)	0242
		04	AE	9F	00018	PUSHAB	4(SP)	0235
00000000G	00		02	FB	00018	CALLS	#2, LIB\$QUAL_FILE_PARSE	
	0A		50	EB	00022	BLBS	STATUS, 1\$	0245
			50	DD	00025	PUSHL	STATUS	0246
00000000G	00		01	FB	00027	CALLS	#1, LIB\$SIGNAL	
				04	0002E	RET		
00000000V	EF		00	FB	0002F	1\$: CALLS	#0, GET_QUALS	0251
	39		50	E9	00036	BLBC	R0, 3\$	
04	AE	020E0000	8F	D0	00039	MOVL	#34471936, DIR_DESC	0258
		08	AE	D4	00041	CLRL	DIR_DESC+4	
		00000000G	00	D4	00044	CLRL	FILE_FAB+24	0259
		04	AE	9F	0004A	2\$: PUSHAB	DIR_DESC	0261
00000000V	EF		01	FB	0004D	CALLS	#1, GET_DIR	
	1B		50	E9	00054	BLBC	R0, 3\$	
		00000000G	00	9F	00057	PUSHAB	SEARCH_ERROR	0264
		00000000V	EF	9F	0005D	PUSHAB	SET_ATTRIBUTES	
		00000000G	00	9F	00063	PUSHAB	FILE_FAB	
00000000G	00		03	FB	00069	CALLS	#3, LIB\$FILE_SCAN	
			D8	11	00070	BRB	2\$	0263
			04	00072	3\$: RET			0271

; Routine Size: 115 bytes, Routine Base: \$CODE\$ + 0000


```
0277 1 ROUTINE get_qual =
0278 1 ++
0279 1
0280 1 This routine interrogates the CLI for qualifiers and values. If an
0281 1 error occurs, it is signaled and control is returned with a value of
0282 1 FALSE.
0283 1
0284 1 --
0285 1 BEGIN
0286 1
0287 1 LOCAL
0288 1     status,
0289 1     desc : $BLOCK[dsc$c_s_bln];           ! General descriptor
0290 1
0291 1 $init_dyndesc(desc);                     ! Make it dynamic
0292 1
0293 1
0294 1 Get the simple flags first.
0295 1
0296 1 setfile$flags[qual_log] = cli$present(%ASCII 'LOG');
0297 1
0298 1
0299 1 /OWNER_UIC
0300 1
0301 1 IF cli$present(%ASCII 'OWNER_UIC')
0302 1 THEN
0303 1 BEGIN
0304 1     setfile$flags[qual_owner] = 1;         ! If present, set bit
0305 1
0306 1 If no value on /OWNER_UIC, then call GETJPI to get this process's UIC.
0307 1
0308 1 IF NOT cli$get_value(%ASCII 'OWNER_UIC', desc)
0309 1 THEN
0310 1 BEGIN
0311 1 LOCAL
0312 1     iosb : VECTOR[4,WORD];
0313 1     status = $GETJPIW(ITMLST = UPLIT(WORD(4,jpi$uic),
0314 1                                     uic_value,
0315 1                                     0,
0316 1                                     0),
0317 1                                     IOSB = iosb);
0318 1
0319 1 IF .status
0320 1 THEN status = .iosb[0];
0321 1 IF NOT .status
0322 1 THEN
0323 1 BEGIN
0324 1     SIGNAL(.status);
0325 1     RETURN false;
0326 1 END;
0327 1 END
0328 1
0329 1 Otherwise, assume a real UIC. Parse the string.
0330 1
0331 1 ELSE
0332 1 BEGIN
0333 1 IF NOT (status = parse_uic(desc, uic_value))
0334 1 THEN
```

```

334 00329 BEGIN
335 00330 SIGNAL(set$ syntax, 1, desc);
336 00331 RETURN false;
337 00332 END;
338 00333 END;
339 00334
340 00335
341 00336
342 00337
343 00338
344 00339
345 00340
346 00341
347 00342
348 00343
349 00344
350 00345
351 00346
352 00347
353 00348
354 00349
355 00350
356 00351
357 00352
358 00353
359 00354
360 00355
361 00356
362 00357
363 00358
364 00359
365 00360
366 00361
367 00362

```

```

BEGIN
SIGNAL(set$ syntax, 1, desc);
RETURN false;
END;

END;

Now for the version limit.
IF cli$present(ASCII 'VERSION_LIMIT')
THEN
BEGIN
setfile$flags[qual_vrsn] = 1;
vrsn_value = 32767;
IF cli$get_value(ASCII 'VERSION_LIMIT', desc)
THEN
BEGIN
IF NOT lib$cvt_dtb(.desc[dsc$w_length],
.desc[dsc$a_pointer],
vrsn_value)
THEN
BEGIN
SIGNAL(set$ syntax, 1, desc);
RETURN false;
END;
IF .vrsn_value LSS 0
OR .vrsn_value GTR 65535
THEN (SIGNAL(set$ valerr); RETURN false);
END;
END;
RETURN true;
END;

```

! Show that /VERSION specified
! Set to the default

```

.PSECT $PLITS,NOWRT,NOEXE,2
00 47 4F 4C 00000 P.AAB: .ASCII \LOG\<0>
010E0003 00004 P.AAA: .LONG 17694723
00000000 00008 .ADDRESS P.AAB
00 00 00 43 49 55 5F 52 45 4E 57 4F 0000C P.AAD: .ASCII \OWNER UIC\<0><0><0>
010E0009 00018 P.AAC: .LONG 17694729
00000000 0001C .ADDRESS P.AAD
00 00 00 43 49 55 5F 52 45 4E 57 4F 00020 P.AAF: .ASCII \OWNER UIC\<0><0><0>
010E0009 0002C P.AAE: .LONG 17694729
00000000 00030 .ADDRESS P.AAF
0304 0004 00034 P.AAG: .WORD 4, 772
00000000G 00038 .ADDRESS UIC_VALUE
00000000 0003C .LONG 0, 0
00 00 54 49 4D 49 4C 5F 4E 4F 49 53 52 45 56 00044 P.AAI: .ASCII \VERSION_LIMIT\<0><0><0>
00 00053
010E000D 00054 P.AAH: .LONG 17694733
00000000 00058 .ADDRESS P.AAI
00 00 54 49 4D 49 4C 5F 4E 4F 49 53 52 45 56 0005C P.AAK: .ASCII \VERSION_LIMIT\<0><0><0>
00 0006B

```

010E000D 0006C P.AAJ: .LONG 17694733
00000000 00070 .ADDRESS P.AAK

.EXTRN SYSSGETJPIW

.PSECT \$CODES,NOWRT,2

00FC 00000 GET_QUALS:

57	00000000G	00	9E	00002	.WORD	Save R2,R3,R4,R5,R6,R7	0272
56	00000000G	00	9E	00009	MOVAB	LIB\$SIGNAL, R7	
55	00000000G	00	9E	00010	MOVAB	CLISGET_VALUE, R6	
54	00000000G	00	9E	00017	MOVAB	VRSN_VALUE, R5	
53	00000000G	00	9E	0001E	MOVAB	SETFILES\$FLAGS, R4	
52	00000000G	00	9E	00025	MOVAB	CLISPRESENT, R3	
5E	00000000G	EF	9E	00025	MOVAB	P.AAA, R2	
08	AE 020E0000	10	C2	0002C	SUBL2	#16, \$P	
	OC	8F	DO	0002F	MOVL	#34471936, DESC	0286
		AE	D4	00037	CLRL	DESC+4	
63		52	DD	0003A	PUSHL	R2	0291
02		01	FB	0003C	CALLS	#1, CLISPRESENT	
	14	50	FO	0003F	INSV	R0, #2, #1, SETFILES\$FLAGS	
63		A2	9F	00044	PUSHAB	P.AAC	0296
42		01	FB	00047	CALLS	#1, CLISPRESENT	
64		50	E9	0004A	BLBC	R0, 3\$	
	08	08	88	0004D	BISB2	#8, SETFILES\$FLAGS	0299
	28	AE	9F	00050	PUSHAB	DESC	0303
66		A2	9F	00053	PUSHAB	P.AAE	
20		02	FB	00056	CALLS	#2, CLISGET_VALUE	
		50	EB	00059	BLBS	R0, 2\$	
		7E	7C	0005C	CLRQ	-(SP)	0312
	08	AE	9F	0005E	PUSHAB	IOSB	
	30	A2	9F	00061	PUSHAB	P.AAG	
		7E	7C	00064	CLRQ	-(SP)	
00000000G	00	7E	D4	00066	CLRL	-(SP)	
06		07	FB	00068	CALLS	#7, SYSSGETJPIW	
50		50	E9	0006F	BLBC	STATUS, 1\$	0313
17		6E	3C	00072	MOVZWL	IOSB, STATUS	0314
		50	E8	00075	BLBS	STATUS, 3\$	0315
		50	DD	00078	PUSHL	STATUS	0318
		67	11	0007A	BRB	7\$	
	00000000G	00	9F	0007C	PUSHAB	UIC VALUE	0327
	OC	AE	9F	00082	PUSHAB	DESC	
00000000G	00	02	FB	00085	CALLS	#2, PARSE UIC	
30		50	E9	0008C	BLBC	STATUS, 4\$	
	50	A2	9F	0008F	PUSHAB	P.AAH	0339
63		01	FB	00092	CALLS	#1, CLISPRESENT	
50		50	E9	00095	BLBC	R0, 8\$	
64		10	88	00098	BISB2	#16, SETFILES\$FLAGS	0342
65	7FFF	8F	3C	0009B	MOVZWL	#32767, VRSN_VALUE	0343
	08	AE	9F	000A0	PUSHAB	DESC	0344
	68	A2	9F	000A3	PUSHAB	P.AAJ	
66		02	FB	000A6	CALLS	#2, CLISGET_VALUE	
3C		50	E9	000A9	BLBC	R0, 8\$	
		55	DD	000AC	PUSHL	R5	0347
	10	AE	DD	000AE	PUSHL	DESC+4	0348
	10	AE	3C	000B1	MOVZWL	DESC, -(SP)	0347
00000000G	00	03	FB	000B5	CALLS	#3, LIB\$CVT_DTB	

SETDIR
V04-000

16-Sep-1984 00:59:34 VAX-11 BLISS-32 V4.0-742
14-Sep-1984 12:09:06 [CLIUTL.SRC]SETDIR.B32;1

Page 14
(6)

10		50	E8	000BC		BLBS	R0, 5\$		
	08	AE	9F	000BF	4\$:	PUSHAB	DESC		0352
		01	DD	000C2		PUSHL	#1		
	007710FA	8F	DD	000C4		PUSHL	#7803130		
67		03	FB	000CA		CALLS	#3, LIB\$SIGNAL		
		1D	11	000CD		BRB	9\$		0353
50		65	D0	000CF	5\$:	MOVL	VRSN_VALUE, R0		0355
		09	19	000D2		BLSS	6\$		
0000FFFF	8F	50	D1	000D4		CMPL	R0, #65535		0356
		0B	15	000DB		BLEQ	8\$		
	007711EA	8F	DD	000DD	6\$:	PUSHL	#7803370		0357
67		01	FB	000E3	7\$:	CALLS	#1, LIB\$SIGNAL		
		04	11	000E6		BRB	9\$		
50		01	D0	000E8	8\$:	MOVL	#1, R0		0361
			04	000EB		RET			
		50	D4	000EC	9\$:	CLRL	R0		0362
			04	000EE		RET			

; Routine Size: 239 bytes, Routine Base: \$CODE\$ + 0073


```
369 0363 1 ROUTINE set_attributes (fab) =
370 0364 1 --
371 0365 1 --
372 0366 1 This is the routine that accesses the directory and sets the
373 0367 1 specified attributes. If an error occurs while attempting to set
374 0368 1 the attributes, a message telling the user is issued, and any other
375 0369 1 directories are processed.
376 0370 1 --
377 0371 1 --
378 0372 1 BEGIN
379 0373 1
380 0374 1 MAP
381 0375 1     fab : REF $BBLOCK;           ! Define the fab
382 0376 1
383 0377 1 LOCAL
384 0378 1     atr : BLOCKVECTOR[3,8,BYTE], ! Attribute control block
385 0379 1     ptr,                          ! Pointer to attribute block
386 0380 1     status,                      ! Status return
387 0381 1     channel : WORD,              ! Channel number
388 0382 1     desc : $BBLOCK[dsc$c s bln], ! General descriptor
389 0383 1     dir_desc : $BBLOCK[dsc$c s bln], ! Descriptor for directory
390 0384 1     string : VECTOR[nam$c maxrss,BYTE], ! String to build directory
391 0385 1     fib : $BBLOCK[fib$c extdata], ! A FIB for the QIO
392 0386 1     header : $BBLOCK[512],       ! The file header
393 0387 1     iosb : VECTOR[4,WORD];       ! I/O status block
394 0388 1
395 0389 1 BIND
396 0390 1     recattr = header[fh2$w_recattr] : $BBLOCK[atr$s_recattr],
397 0391 1
398 0392 1     nam = .fab[fab$l_nam] : $BBLOCK; ! Define the name block
399 0393 1
400 0394 1 IF .setfile$flags[qual_quit]
401 0395 1 THEN RETURN true;
402 0396 1
403 0397 1 !
404 0398 1 ! Build a descriptor to point to the directory.
405 0399 1 !
406 0400 1 CHSMOVE(.nam[nam$b_rsl],          ! Move this many characters
407 0401 1     .nam[nam$l_rsa],              ! From the nam
408 0402 1     string);                     ! To the temp string
409 0403 1 dir_desc[dsc$a_pointer] = string; ! Point to the string
410 0404 1 dir_desc[dsc$w_length] = .nam[nam$b_rsl]; ! Fill in the size
411 0405 1 !
412 0406 1 ! Move the final bracket over, to get the name of the final directory
413 0407 1 !
414 0408 1 string[.nam[nam$b_dev]
415 0409 1     + .nam[nam$b_dir]
416 0410 1     + .nam[nam$b_name]] = .string[.nam[nam$b_dev]
417 0411 1     + .nam[nam$b_dir]
418 0412 1     - 1];
419 0413 1 string[.nam[nam$b_dev]
420 0414 1     + .nam[nam$b_dir] - 1] = '.'; ! Replace bracket with dot
421 0415 1 dir_desc[dsc$w_length] = .nam[nam$b_dev]
422 0416 1     + .nam[nam$b_dir]
423 0417 1     + .nam[nam$b_name] + 1; ! Update string length
424 0418 1 !
425 0419 1 ! If the directory looks like [000000.something], get rid of the leading
```

```
426 0420 2 | zeroes and dot.
427 0421
428 0422
429 0423 IF NOT CH$FAIL(status = CH$FIND_SUB(.dir_desc[dsc$w_length],
430 0424 .dir_desc[dsc$a_pointer],
431 0425 6 UPLIT('000000'))
432 0426
433 0427 THEN
434 0428 BEGIN
435 0429 CH$MOVE(.dir_desc[dsc$w_length] - (.status + 7 - .dir_desc[dsc$a_pointer]),
436 0430 .status + 7,
437 0431 .status);
438 0432 dir_desc[dsc$w_length] = .dir_desc[dsc$w_length] - 7;
439 0433 END;
440 0434
441 0435
442 0436 | Check to see if this directory file matches the requested qualifiers
443 0437
444 0438 status = lib$qual_file_match(context, | Call the common qualifier routine
445 0439 fab, | fab for directory
446 0440 0
447 0441 %ASCID 'Modify !AS? [N]: ',
448 0442 %REF(dir_desc));
449 0443
450 0444 IF NOT .status | If the status is false, check
451 0445 THEN | what kind of "error"
452 0446 BEGIN
453 0447 IF .status EQL lib$_filfailmat | If no match, return
454 0448 THEN RETURN true
455 0449 ELSE IF .status EQL lib$_quipro | If quit-right now,
456 0450 THEN
457 0451 BEGIN
458 0452 setfile$flags[qual_quit] = 1; | set a flag
459 0453 RETURN true; | and leave
460 0454 END
461 0455 ELSE | For any other error,
462 0456 BEGIN
463 0457 SIGNAL(.status); | signal it and
464 0458 RETURN true; | go away
465 0459 END;
466 0460 END;
467 0461
468 0462
469 0463 | Assign a channel to the file's device
470 0464
471 0465 desc[dsc$w_length] = .nam[nam$b dev]; | Set up the descriptor
472 0466 desc[dsc$a_pointer] = .nam[nam$_dev]; | to point to the device name
473 0467
474 p 0468 IF NOT (status = $ASSIGN(
475 0469 DEVNAM = desc
476 0470 (CHAN = channe()))
477 0471 THEN
478 0472 BEGIN
479 0473 file_error( set$openin,
480 0474 .status, .fab); | Tell user why the assign failed
481 0475 RETURN true; | And continue with other files
482 0476 END;
```



```
540 BEGIN
541
542
543 Change the version limit for the directory
544
545     atr[0,atr$w_type] = atr$c_recattr;
546     atr[0,atr$w_size] = atr$s_recattr;
547     atr[0,atr$l_addr] = recattr;
548     recattr[fat$w_versions] = .vrsn_value;
549     ptr = .ptr + 1; ! Update the pointer
550 END;
551
552 If .setfile$flags[qual_owner]
553 THEN
554     BEGIN
555 Change the owner uic
556
557     atr[.ptr,atr$w_type] = atr$c_uic;
558     atr[.ptr,atr$w_size] = atr$s_uic;
559     atr[.ptr,atr$l_addr] = uic_value;
560     ptr = .ptr + 1; ! Update the pointer
561 END;
562
563
564 Look to see if the pointer has been changed. If yes, modify the header.
565
566 IF .ptr NEQ 0
567 THEN
568     BEGIN
569         atr[.ptr,0,0,32,0] = 0; ! Put a zero at end of list
570         status = $QIOW( CHAN = .channel, ! Make the modifications
571             FUNC = IOS MODIFY,
572             IOSB = iosb,
573             P1 = desc,
574             P5 = atr);
575         IF .status THEN status = .iosb[0];
576         IF NOT .status ! If the modify failed, tell user
577             THEN file_error(set$writeerr,.status,.fab)
578         ELSE
579             IF .setfile$flags[qual_log] ! If /LOG, tell user
580                 THEN SIGNAL(set$modified,1,dir_desc);
581             $QIOW(CHAN = .channel, ! Deaccess the file
582                 FUNC = IOS DEACCESS,
583                 IOSB = iosb,
584                 P1 = desc);
585             END; ! End of header modify
586         END; ! End of qualifier scan
587
588 Deassign the channel
589
590 IF NOT (status = $DASSGN(CHAN = .channel))
591 THEN file_error(set$closeerr,.status,.fab);
592 END;
593
594 RETURN true;
595 END;
```



```
.PSECT $SPLITS,NOWRT,NOEXE,2

5D 4E 5B 20 3F 53 41 21 20 79 66 69 64 6F 4D 00074 P.AAL: .ASCII \000000\<0><0>
                                00 00 00 20 3A 0007C P.AAN: .ASCII \Modify !AS? [N]: \<0><0><0>
                                010E0011 00090 P.AAM: .LONG 17694737
                                00000000' 00094 .ADDRESS P.AAN

                                .EXTRN SYS$ASSIGN, SYS$QIOW
                                .EXTRN SYS$DASSGN

.PSECT $CODE$,NOWRT,2

                                OFFC 00000 SET_ATTRIBUTES:
                                .WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
                                MOVAB SYS$QIOW, R11
                                MOVAB LIB$SIGNAL, R10
                                MOVAB SETFILES$FLAGS, R9
                                MOVAB -856(SP), SP
                                MOVL FAB, R8
                                MOVL 40(R8), R6
                                BBC #1, SETFILES$FLAGS, 1$
                                BRW 19$
                                MOVZBL 3(R6), R0
                                MOVCL R0, @4(R6), STRING
                                MOVAB STRING, DIR_DESC+4
                                MOVZBW 3(R6), DIR_DESC
                                MOVZBL 57(R6), R0
                                MOVZBL 58(R6), R1
                                ADDL2 R1, R0
                                MOVZBL 59(R6), R1
                                ADDL2 R0, R1
                                MOVB STRING-1(R0), STRING[R1]
                                MOVB #46, STRING-1(R0)
                                ADDW3 #1, R1, DIR_DESC
                                MATCHC #6, P.AAL, DIR_DESC, @DIR_DESC+4
                                BEQL 2$
                                MOVL #6, R3
                                SUBL2 #2, R3
                                MOVAL -(R3), STATUS
                                BEQL 3$
                                SURL3 STATUS, DIR_DESC+4, R0
                                MOVZWL DIR_DESC, R1
                                ADDL2 R1, -R0
                                SUBL2 #7, R0
                                MOVL STATUS, R3
                                MOVCL R0, 7(R3), (STATUS)
                                SUBW2 #7, DIR_DESC
                                MOVAB DIR_DESC, (SP)
                                PUSHL SP
                                PUSHAB P.AAM
                                CLRL -(SP)
                                PUSHL R8
                                PUSHAB CONTEXT

                                0363
                                0392
                                0394
                                0400
                                0403
                                0404
                                0409
                                0410
                                0414
                                0417
                                0425
                                0429
                                0430
                                0431
                                0432
                                0442
                                0440
                                0438
                                0439
                                0438
```

5D 4E 5B 20 3F 53 41 21 20 79 66 69 64 6F 4D 00074 P.AAL: .ASCII \000000\<0><0>
00 00 00 20 3A 0007C P.AAN: .ASCII \Modify !AS? [N]: \<0><0><0>
010E0011 00090 P.AAM: .LONG 17694737
00000000' 00094 .ADDRESS P.AAN

.EXTRN SYS\$ASSIGN, SYS\$QIOW
.EXTRN SYS\$DASSGN

.PSECT \$CODE\$,NOWRT,2

OFFC 00000 SET_ATTRIBUTES:

.WORD Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11
MOVAB SYS\$QIOW, R11
MOVAB LIB\$SIGNAL, R10
MOVAB SETFILES\$FLAGS, R9
MOVAB -856(SP), SP
MOVL FAB, R8
MOVL 40(R8), R6
BBC #1, SETFILES\$FLAGS, 1\$
BRW 19\$
MOVZBL 3(R6), R0
MOVCL R0, @4(R6), STRING
MOVAB STRING, DIR_DESC+4
MOVZBW 3(R6), DIR_DESC
MOVZBL 57(R6), R0
MOVZBL 58(R6), R1
ADDL2 R1, R0
MOVZBL 59(R6), R1
ADDL2 R0, R1
MOVB STRING-1(R0), STRING[R1]
MOVB #46, STRING-1(R0)
ADDW3 #1, R1, DIR_DESC
MATCHC #6, P.AAL, DIR_DESC, @DIR_DESC+4
BEQL 2\$
MOVL #6, R3
SUBL2 #2, R3
MOVAL -(R3), STATUS
BEQL 3\$
SURL3 STATUS, DIR_DESC+4, R0
MOVZWL DIR_DESC, R1
ADDL2 R1, -R0
SUBL2 #7, R0
MOVL STATUS, R3
MOVCL R0, 7(R3), (STATUS)
SUBW2 #7, DIR_DESC
MOVAB DIR_DESC, (SP)
PUSHL SP
PUSHAB P.AAM
CLRL -(SP)
PUSHL R8
PUSHAB CONTEXT

0363
0392
0394
0400
0403
0404
0409
0410
0414
0417
0425
0429
0430
0431
0432
0442
0440
0438
0439
0438

```

0 2
16-Sep-1984 00:59:34 VAX-11 Bliss-32 V4.0-742
14-Sep-1984 12:09:06 [CLIUTL.SRC]SETDIR.B32:1

```

Page 20
(7)

Address	Hex	Assembly	Comment	Symbol
00000000G	00	05 FB 000B0	CALLS	#5, LIB\$QUAL_FILE_MATCH
	57	50 D0 000B7	MOVL	R0, STATUS
	1F	57 E8 000BA	BLBS	STATUS, 6\$
00000000G	8F	57 D1 000BD	CMPL	STATUS, #LIB\$FILFAIMAT
		13 13 000C4	BEQL	5\$
00000000G	8F	57 D1 000C6	CMPL	STATUS, #LIB\$QUIPRO
		05 12 000CD	BNEQ	4\$
	69	02 88 000CF	BISB2	#2, SETFILES\$FLAGS
		05 11 000D2	BRB	5\$
		57 DD 000D4	PUSHL	STATUS
	6A	01 FB 000D6	CALLS	#1, LIB\$SIGNAL
		018D 31 000D9	BRW	19\$
E0	AD	A6 9B 000DC	MOVZBW	57(R6), DESC
E4	AD	A6 D0 000E1	MOVL	68(R6), DESC+4
		7E 7C 000E6	CLRQ	-(SP)
		OC AE 9F 000E8	PUSHAB	CHANNEL
	E0	AD 9F 000EB	PUSHAB	DESC
00000000G	00	04 FB 000EE	CALLS	#4, SYSS\$ASSIGN
	57	50 D0 000F5	MOVL	R0, STATUS
	0B	57 E8 000F8	BLBS	STATUS, 7\$
	7E	57 7D 000FB	MOVQ	STATUS, -(SP)
		8F DD 000FE	PUSHL	#7803034
		69 11 00104	BRB	9\$
		20 B0 00106	MOVW	#32, DESC
E0	AD	CD 9E 0010A	MOVAB	FIB, DESC+4
E4	AD	00 2C 00110	MOVCS	#0, (SP), #0, #32, FIB
	6E	CD 00115		
		8F 3C 00118	MOVZWL	#1281, FIB
		24 A6 D0 0011F	MOVL	36(R6), FIB+4
		28 A6 B0 00125	MOVW	40(R6), FIB+8
FEB8	CD	8F D0 0012B	MOVL	#655872, ATR
FEB8	CD	AE 9E 00133	MOVAB	HEADER, ATR+4
FEC0	CD	AD D4 00138	CLRL	ATR+8
E8	AD	7E D4 0013B	CLRL	-(SP)
EC	AD	AD 9F 0013D	PUSHAB	ATR
		7E 7C 00140	CLRQ	-(SP)
		7E D4 00142	CLRL	-(SP)
		AD 9F 00144	PUSHAB	DESC
		7E 7C 00147	CLRQ	-(SP)
		28 AE 9F 00149	PUSHAB	IOSB
		72 8F 9A 0014C	MOVZBL	#114, -(SP)
		2C AE 3C 00150	MOVZWL	CHANNEL, -(SP)
		7E D4 00154	CLRL	-(SP)
		OC FB 00156	CALLS	#12, SYSS\$QIOW
		50 D0 00159	MOVL	R0, STATUS
		57 E9 0015C	BLBC	STATUS, 8\$
		AE 3C 0015F	MOVZWL	IOSB, STATUS
		57 E8 00163	BLBS	STATUS, 10\$
		57 7D 00166	MOVQ	STATUS, -(SP)
		8F DD 00169	PUSHL	#SET\$_READERR
		00F0 31 0016F	BRW	18\$
		AE 91 00172	CMPB	HEADER+7, #2
		26 12 00176	BNEQ	11\$
		05 E0 00178	BBS	#5, HEADER+53, 11\$
		A6 9B 0017D	MOVZBW	3(R6), DESC
		A6 D0 00182	MOVL	4(R6), DESC+4
		8F DD 00187	PUSHL	#SET\$_NOTDIR

		EO	AD	9F	0018D	PUSHAB	DESC		
			01	DD	00190	PUSHL	#1		
		00000000G	8F	DD	00192	PUSHL	#SET\$ WRITEERR		
	6A		04	FB	00198	CALLS	#4, LIB\$SIGNAL		
			00AA	31	0019B	BRW	17\$		0513
			50	D4	0019E	CLRL	PTR		0530
17		69	04	E1	001A0	BBC	#4, SETFILES\$FLAGS, 12\$		0532
	E8	AD	00040020	8F	D0	MOVL	#262176, ATR		0540
	EC	AD	24	AE	9E	MOVAB	RECATTR, ATR+4		0541
	42	AE	00000000G	00	B0	MOVW	VRSN_VALUE, RECATTR+30		0542
				50	D6	INCL	PTR		0543
18		69		03	E1	BBC	#3, SETFILES\$FLAGS, 13\$		0546
		EA	AD40	7F	001BF	PUSHAQ	ATR+2[PTR]		0552
		9E		15	B0	MOVW	#21, @ (SP)+		
		E8	AD40	7F	001C6	PUSHAQ	ATR[PTR]		0553
		9E		04	B0	MOVW	#4, @ (SP)+		
		EC	AD40	7F	001CD	PUSHAQ	ATR+4[PTR]		0554
		9E	00000000G	00	9E	MOVAB	UIC_VALUE, @ (SP)+		
				50	D6	INCL	PTR		0555
				6C	13	BEQL	17\$		0561
		E8	AD40	7F	001DC	PUSHAQ	ATR[PTR]		0564
				9E	D4	CLRL	@ (SP)+		
				7E	D4	CLRL	-(SP)		0569
		E8	AD	9F	001E4	PUSHAB	ATR		
				7E	7C	CLRQ	-(SP)		
				7E	D4	CLRL	-(SP)		
		EO	AD	9F	001EB	PUSHAB	DESC		
				7E	7C	CLRQ	-(SP)		
		28	AE	9F	001F0	PUSHAB	IOSB		
				36	DD	PUSHL	#54		
	7E	2C	AE	3C	001F5	MOVZWL	CHANNEL, -(SP)		
				7E	D4	CLRL	-(SP)		
	6B		0C	FB	001FB	CALLS	#12, SYS\$QIOW		
	57		50	D0	001FE	MOVL	R0, STATUS		
	07		57	E9	00201	BLBC	STATUS, 14\$		0570
	57	08	AE	3C	00204	MOVZWL	IOSB, STATUS		
	12		57	E8	00208	BLBS	STATUS, 15\$		0571
	7E		57	7D	0020B	MOVQ	STATUS, -(SP)		0572
		00000000G	00	8F	DD	PUSHL	#SET\$ WRITEERR		
				03	FB	CALLS	#3, FILE_ERROR		
				12	11	BRB	16\$		
0E		69		02	E1	BBC	#2, SETFILES\$FLAGS, 16\$		0574
				AD	9F	PUSHAB	DIR_DESC		0575
				01	DD	PUSHL	#1		
		00000000G	8F	DD	00226	PUSHL	#SET\$ MODIFIED		
	6A		03	FB	0022C	CALLS	#3, LIB\$SIGNAL		
			7E	7C	0022F	CLRQ	-(SP)		0579
			7E	7C	00231	CLRQ	-(SP)		
			7E	D4	00233	CLRL	-(SP)		
		EO	AD	9F	00235	PUSHAB	DESC		
				7E	7C	CLRQ	-(SP)		
		28	AE	9F	0023A	PUSHAB	IOSB		
				34	DD	PUSHL	#52		
	7E	2C	AE	3C	0023F	MOVZWL	CHANNEL, -(SP)		
				7E	D4	CLRL	-(SP)		
	6B		0C	FB	00245	CALLS	#12, SYS\$QIOW		
	7E		04	AE	3C	MOVZWL	CHANNEL, -(SP)		0585

SETDIR
V04-000

D 2
16-Sep-1984 00:59:34
14-Sep-1984 12:09:06

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETDIR.B32;1

Page 22
(7)

00000000G	00	01	FB	0024C	CALLS	#1, SYSSDASSGN
	57	50	DO	00253	MOVL	R0, STATUS
	10	57	EB	00256	BLBS	STATUS, 198
	7E	57	7D	00259	MOVQ	STATUS, -(SP)
		8F	DD	0025C	PUSHL	#SETS, CLOSEERR
00000000G	00	03	FB	00262	CALLS	#3, FILE_ERROR
	50	01	DO	00269	MOVL	#1, R0
		04	0026C	RET		

0586

0589

0590

; Routine Size: 621 bytes, Routine Base: \$CODE\$ + 0162


```
598 0591 1 ROUTINE get_dir (dir_desc) =
599 0592 1 ++
600 0593 1
601 0594 1 This routine is called to get each directory spec from the command line.
602 0595 1 When there are no more directories, a value of FALSE is returned.
603 0596 1
604 0597 1 --
605 0598 1 BEGIN
606 0599 1
607 0600 1 MAP dir_desc : REF $BBLOCK;
608 0601 1
609 0602 1 LOCAL
610 0603 1     desc : $BBLOCK[dsc$_s_bln],           ! General descriptor
611 0604 1     endchar : BYTE,                       ! End-of-directory character
612 0605 1     eos,                                   ! End of string
613 0606 1     ptr,                                   ! Pointer in string
614 0607 1     str_ptr,                             ! Pointer to rest of string
615 0608 1     str_len,                             ! Length of rest of string
616 0609 1     temp_string : VECTOR[nam$_maxrss,BYTE], ! The temporary string
617 0610 1     temp,
618 0611 1     status;
619 0612 1
620 0613 1 IF .setfile$flags[qual_quit]
621 0614 1 THEN RETURN false;
622 0615 1
623 0616 1
624 0617 1 Check here to see if a trailing ellipse is being treated. If so,
625 0618 1 then FILE_FAB[FAB$_CTX] will be set to TRUE, and there's no need
626 0619 1 to search and see if such a trailing ellipse is present. However,
627 0620 1 if the value is set to FALSE, then get a new directory spec.
628 0621 1
629 0622 1 IF NOT .file_fab[fab$_ctx]                ! If not processing an ellipse
630 0623 1 THEN                                     ! then get the next directory
631 0624 1 BEGIN
632 0625 1     IF NOT cli$get_value(%ASCII 'DIRECTORY', .dir_desc)
633 0626 1     THEN RETURN false;
634 0627 1
635 0628 1     file_fab[fab$_fna] = .dir_desc[dsc$_a_pointer];
636 0629 1     file_fab[fab$_fns] = .dir_desc[dsc$_w_length];
637 0630 1
638 0631 1
639 0632 1 Since this is a new entry, it must be checked for a trailing ellipse.
640 0633 1
641 0634 1 CHSMOVE( .dir_desc[dsc$_w_length],      ! Move this many chars
642 0635 1         .dir_desc[dsc$_a_pointer],    ! From the CLI area
643 0636 1         temp_string);                  ! To the temp string
644 0637 1 str_ptr = temp_string;                ! Set up pointer
645 0638 1 str_len = .dir_desc[dsc$_w_length];    ! and length.
646 0639 1
647 0640 1 Look for ellipses.
648 0641 1
649 0642 1 WHILE NOT CH$FAIL(temp = CH$FIND_SUB(.str_len, .str_ptr, 3, UPLIT('...')))
650 0643 1 DO
651 0644 1 BEGIN
652 0645 1     str_ptr = .temp + 3;                ! Update pointer
653 0646 1     str_len = .str_len - (.temp - .str_ptr) - 3;
654 0647 1 END;
```

```
655 0648 .....
656 0649 .....
657 0650 ..... After the final ellipse, check to see if it is at the end of the
658 0651 ..... directory specification. If so, then change the context field of
659 0652 ..... the fab, and insert an end bracket at the beginning of the ellipse.
660 0653 .....
661 0654 ..... IF (.str_ptr EQL temp_string + .file_fab[fab$b_fns] - 1)
662 0655 ..... THEN
663 0656 ..... BEGIN
664 0657 .....     file_fab[fab$l_ctx] = true;           ! Show that there's a trailing ellipse
665 0658 .....     CH$WCHAR(..str_ptr, .str_ptr - 3);   ! Put the end bracket in place
666 0659 .....     file_fab[fab$l_fna] = temp_string;   ! Set up FAB fields
667 0660 .....     file_fab[fab$b_fns] = .str_ptr - 3 - temp_string + 1;
668 0661 ..... END;
669 0662 ..... END
670 0663 .....
671 0664 ..... If here, then the trailing ellipse has been processed, and this is the
672 0665 ..... second time thru. Restore the original file name.
673 0666 ..... ELSE
674 0667 ..... BEGIN
675 0668 .....     file_fab[fab$l_fna] = .dir_desc[dsc$a_pointer];   ! Original filename
676 0669 .....     file_fab[fab$b_fns] = .dir_desc[dsc$w_length];   ! Original length
677 0670 .....     file_fab[fab$l_ctx] = false;                     ! Ellipse processed
678 0671 ..... END;
679 0672 .....
680 0673 ..... Parse the input string
681 0674 .....
682 0675 ..... $NAM_INIT (NAM = file_rlf);           ! Re-init the RLF
683 0676 ..... IF (.file_nam[nam$b_dev] NEQ 0)       ! If a device was
684 0677 ..... THEN                                     ! specified, then
685 0678 ..... BEGIN
686 0679 .....     file_fab[fab$l_dna] = .file_nam[nam$l_dev];
687 0680 .....     file_fab[fab$b_dns] = .file_nam[nam$b_dev];
688 0681 ..... END;
689 0682 ..... ! Make device sticky
690 0683 ..... IF NOT (status = $PARSE(FAB = file_fab))
691 0684 ..... THEN
692 0685 ..... BEGIN
693 0686 .....     desc[dsc$w_length] = .file_fab[fab$b_fns];
694 0687 .....     desc[dsc$a_pointer] = .file_fab[fab$l_fna];
695 0688 .....     SIGNAL_STOP(set$syntax,
696 0689 .....         1, desc, .status);
697 0690 ..... END;
698 0691 .....
699 0692 .....
700 0693 ..... Check the parsed string for legality, i.e. nothing after the directory
701 0694 .....
702 0695 ..... IF (.file_nam[nam$b_name] NEQ 0 OR
703 0696 .....     .file_nam[nam$b_type] NEQ 1 OR
704 0697 .....     .file_nam[nam$b_ver] NEQ 1 )
705 0698 ..... THEN
706 0699 ..... BEGIN
707 0700 .....     desc[dsc$w_length] = .file_fab[fab$b_fns];
708 0701 .....     desc[dsc$a_pointer] = .file_fab[fab$l_fna];
709 0702 .....     SIGNAL_STOP(set$syntax,
710 0703 .....         1, desc, ss$b_badirectory);
711 0704 ..... END;
```

```
0705 2
0706 2 Check if a search list is used. If so SIGNAL, as search lists are not
0707 2 allowed (for devices) in this application.
0708 2
0709 2 IF .file_nam[nam$u_search_list]
0710 2 THEN
0711 2     SIGNAL_STOP(set$_filspcrch, 1, .dir_desc, set$_nosrchlst);
0712 2
0713 2 Determine what the directory terminator character was, and save it.
0714 2
0715 2 endchar = .(.file_nam[nam$l_dir] + .file_nam[nam$b_dir] - 1);
0716 2
0717 2
0718 2 The directory string must now be analyzed and manipulated so that the
0719 2 final directory entry becomes a file. First, initialize some pointers.
0720 2
0721 2 desc[dsc$w_length] = .file_nam[nam$b_esl] - 2;
0722 2 desc[dsc$a_pointer] = .file_nam[nam$_esa];
0723 2 str_ptr = .desc[dsc$a_pointer];
0724 2 str_len = .desc[dsc$w_length];
0725 2 ptr = 0;
0726 2 eos = .desc[dsc$a_pointer] + .desc[dsc$w_length] - 1;
0727 2
0728 2
0729 2 Look for wildcard ellipses
0730 2
0731 2 WHILE NOT CH$FAIL(temp = CH$FIND_SUB(.str_len, .str_ptr, 3, uplit('...')))
0732 2 DO
0733 2     BEGIN
0734 2
0735 2 Make PTR point to the beginning of the "...", and advance the string
0736 2 pointer to the character just past the "...
0737 2
0738 2     ptr = .temp;
0739 2     str_len = .str_len - (.temp - .str_ptr) - 3;
0740 2     str_ptr = .temp + 3;
0741 2     END;
0742 2
0743 2
0744 2 If there was any occurrence of "...", point just past it.
0745 2
0746 2 IF .ptr NEQ 0 THEN ptr = .ptr + 3;
0747 2
0748 2
0749 2 Find the last directory in the specification
0750 2
0751 2 WHILE NOT CH$FAIL( temp = CH$FIND_CH(.str_len, .str_ptr, '.'))
0752 2 DO
0753 2     BEGIN
0754 2
0755 2 Make PTR point to the ".", and advance the string pointer to
0756 2 the first character after the "."
0757 2
0758 2     ptr = .temp;
0759 2     str_len = .str_len - (.temp - .str_ptr) - 1;
0760 2     str_ptr = .temp + 1;
0761 2     END;
```

```
769 0762
770 0763 IF .ptr NEQ 0
771 0764 THEN
772 0765 BEGIN
773 0766
774 0767 If here, then either a trailing ellipse, or a final sub-directory
775 0768 was specified. If the pointer is at the bracket, then there is a
776 0769 trailing ellipse, in which case only a '*' is required.
777 0770
778 0771 IF .ptr EQL .eos
779 0772 THEN
780 0773 BEGIN
781 0774 CHSA WCHAR('*', ptr); ! Stick an asterisk after the bracket.
782 0775 ptr = .ptr + 1; ! Adjust the pointer.
783 0776 END
784 0777
785 0778 If the pointer is inside the bracket, then the last directory name
786 0779 must be moved out of the brackets.
787 0780
788 0781 ELSE
789 0782 BEGIN
790 0783
791 0784 Check to see if the directory is [main.sub] or [main...sub]
792 0785
793 0786 IF .ptr EQLU .str_ptr
794 0787 THEN
795 0788 BEGIN ! [main...sub] form
796 0789 str_len = .eos - .ptr;
797 0790 CHSMOVE(.str_len, .ptr, .ptr+1);
798 0791 CHSWCHAR(.endchar, .ptr);
799 0792 ptr = .ptr + .str_len + 1;
800 0793 END ! end of [main...sub] processing
801 0794 ELSE
802 0795 BEGIN ! [main.sub] form
803 0796 str_len = .eos - .str_ptr;
804 0797 CHSWCHAR A(.endchar, ptr);
805 0798 ptr = .ptr + .str_len;
806 0799 END ! end of [main.sub] processing
807 0800 END;
808 0801 END ! End of non-zero pointer stuff
809 0802 ELSE
810 0803 BEGIN
811 0804
812 0805 If the pointer is still zero, then there is either a wildcard, a main
813 0806 directory, or a [g.m] directory. In all such cases, a main directory
814 0807 of [000000] must be fabricated.
815 0808
816 0809 status = CHSFIND_CH(.str_len, .str_ptr, ','); ! Save for later
817 0810
818 0811 Move the string out seven spaces and insert "000000]"
819 0812
820 0813 str_ptr = .desc[dsc$a_pointer] + .file_nam[nam$b dev] + 1;
821 0814 temp = CHSMOVE(.eos - .str_ptr, .str_ptr, .str_ptr + 7);
822 0815 str_ptr = CHSMOVE(6, UPLITT'000000', .str_ptr);
823 0816 CHSMOVE(1, endchar, .str_ptr);
824 0817
825 0818 If no comma was found, then all that is required is to update the
```



```
0819 pointer.  
0820  
0821 IF CH$FAIL(.status) THEN ptr = .temp  
0822  
0823 Otherwise, it's a [g,m] directory. Convert it.  
0824  
0825 ELSE  
0826 BEGIN  
0827  
0828 LOCAL tparse_block : $BLOCK[tpa$length0]; ! Define a TPARSE block  
0829  
0830 CH$FILL(0, tpa$length0, tparse_block); ! Zero it.  
0831 tparse_block[tpa$l_count] = tpa$l_count0; ! Fill in size  
0832  
0833 tparse_block[tpa$l_stringcnt] = .eos - .str_ptr;  
0834 tparse_block[tpa$l_stringptr] = .str_ptr + 7;  
0835 IF NOT (status = lib$tparse(tparse_block,  
0836 dir_state,  
0837 dir_keys))  
0838 THEN SIGNAL_STOP(.status)  
0839 ELSE  
0840 BEGIN  
0841 LOCAL temp_desc : $BLOCK[dsc$sc_s_bln];  
0842 temp_desc[dsc$w_length] = 6;  
0843 temp_desc[dsc$a_pointer] = .str_ptr + 7;  
0844 IF NOT (status = SYS$FAO($descriptor('!2(30W)'),  
0845 temp_desc,  
0846 temp_desc,  
0847 .group,  
0848 .member))  
0849 THEN SIGNAL_STOP(.status)  
0850 ELSE ptr = .str_ptr + 14;  
0851 END;  
0852 END;  
0853 END;  
0854 ptr = CH$MOVE(4, UPLIT('.DIR'), .ptr);  
0855 file_fab[fab$b_fns] = .ptr - .desc[dsc$a_pointer];  
0856 file_fab[fab$l_fna] = .desc[dsc$a_pointer];  
0857  
0858 RETURN true;  
0859 1 END;
```

```
00 00 00 59 52 4F 54 43 45 52 49 44 00098 P.AAP: .ASCII \DIRECTORY\<0><0><0>  
010E0009 000A4 P.AAO: .LONG 17694729  
00000000 000A8 .ADDRESS P.AAP  
00 2E 2E 2E 000AC P.AAQ: .ASCII \...\<0>  
00 2E 2E 2E 000B0 P.AAR: .ASCII \...\<0>  
00 00 30 30 30 30 30 30 000B4 P.AAS: .ASCII \000000\<0><0>  
29 57 4F 33 28 32 21 000BC P.AAU: .ASCII \!2(30W)\  
000C3 .BLKB 1  
00000007 000C4 P.AAT: .LONG 7  
00000000 000C8 .ADDRESS P.AAU  
52 49 44 2E 000CC P.AAV: .ASCII \.DIR\
```

.PSECT \$SPLITS,NOWRT,NOEXE,2

		SRMS_PTR=		FILE_RLF						
		.EXTRN		SYSSPARSE						
		.PSECT		SCODES, NOWRT, 2						
			OFFC	00000	GET_DIR: .WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	0591			
		5E	FEC4	CE	9E	00002	MOVAB	-316(SP), SP		
03	00000000G	00		01	E1	00007	BBC	#1, SETFILE\$FLAGS, 28	0613	
		56	04	02F8	31	0000F	BRW	29\$		
		03	00000000G	AC	D0	00012	2\$: MOVL	DIR_DESC, R6	0625	
				00	E9	00016	BLBC	FILE_FAB+24, 3\$	0622	
				008A	31	0001D	BRW	7\$		
				56	DD	00020	3\$: PUSHL	R6	0625	
			00000000'	EF	9F	00022	PUSHAB	P.AAO		
	00000000G	00		02	FB	00028	CALLS	#2, CLISGET_VALUE		
		DD		50	E9	0002F	BLBC	R0, 1\$		
	00000000G	00	04	A6	D0	00032	MOVL	4(R6), FILE_FAB+44	0628	
	00000000G	00		66	90	0003A	MOVB	(R6), FILE_FAB+52	0629	
34	AE	04		66	28	00041	MOVC3	(R6), #4(R6), TEMP_STRING	0634	
		5A	34	AE	9E	00047	MOVAB	TEMP_STRING, STR_PTR	0637	
		57		66	3C	0004B	MOVZWL	(R6), STR_LEN	0638	
6A		57	00000000'	03	39	0004E	4\$: MATCHC	#3, P.AAO, STR_LEN, (STR_PTR)	0642	
				03	13	00057	BEQL	5\$		
		53		03	D0	00059	MOVL	#3, R3		
		5B		53	D7	0005C	5\$: DECL	R3		
				73	3E	0005E	MOVAB	-(R3), TEMP		
				0F	13	00061	BEQL	6\$		
		5A	03	AB	9E	00063	MOVAB	3(R11), STR_PTR	0645	
53		5A		5B	C3	00067	SUBL3	TEMP, STR_PTR, R3	0646	
		57	FD	A347	9E	0006B	MOVAB	-3(R3)[STR_LEN], STR_LEN		
				DC	11	00070	BRB	4\$	0642	
		50	00000000G	00	9A	00072	6\$: MOVZBL	FILE_FAB+52, R0	0653	
		50		33	AE40	9E	00079	MOVAB	TEMP_STRING-1[R0], R0	
		50		5A	D1	0007E	CMPL	STR_PTR, R0		
				3C	12	00081	BNEQ	8\$		
	00000000G	00		01	D0	00083	MOVL	#1, FILE_FAB+24	0656	
		51	FD	AA	9E	0008A	MOVAB	-3(STR_PTR), R1	0657	
		61		6A	90	0008E	MOVB	(STR_PTR), (R1)		
	00000000G	00	34	AE	9E	00091	MOVAB	TEMP_STRING, FILE_FAB+44	0658	
		50	34	AE	9E	00099	MOVAB	TEMP_STRING, R0	0659	
		50		51	C2	0009D	SUBL2	R1, R0		
	00000000G	00		50	83	000A0	SUBB3	R0, #1, FILE_FAB+52		
		01		15	11	000AB	BRB	8\$	0622	
	00000000G	00	04	A6	D0	000AA	7\$: MOVL	4(R6), FILE_FAB+44	0668	
	00000000G	00		66	90	000B2	MOVB	(R6), FILE_FAB+52	0669	
			00000000G	00	D4	000B9	CLRL	FILE_FAB+24	0670	
0060	8F	00		00	2C	000BF	8\$: MOVC5	#0, (SP), #0, #96, SRMS_PTR	0676	
				EF		000C6				
	00000000'	EF	6002	8F	80	000CB	MOVW	#24578, SRMS_PTR		
		50	00000000G	00	9A	000D4	MOVZBL	FILE_NAM+57, R0	0677	
				12	13	000DB	BEQL	9\$		
	00000000G	00	00000000G	00	D0	000DD	MOVL	FILE_NAM+68, FILE_FAB+48	0680	
	00000000G	00		50	90	000EB	MOVB	R0, FILE_FAB+53	0681	
			00000000G	00	9F	000EF	9\$: PUSHAB	FILE_FAB	0683	
	00000000G	00		01	FB	000F5	CALLS	#1, SYSSPARSE		
		6E		50	D0	000FC	MOVL	R0, STATUS		

	24	6E	E8	000FF	BLBS	STATUS, 108	
F8	AD	00000000G	00	9B	00102	MOVZBW	FILE_FAB+52, DESC
FC	AD	00000000G	00	D0	0010A	MOVL	FILE_FAB+44, DESC+4
			6E	DD	00112	PUSHL	STATUS
	F8		AD	9F	00114	PUSHAB	DESC
			01	DD	00117	PUSHL	#1
	007710FA		8F	DD	00119	PUSHL	#7803130
00000000G	00		04	FB	0011F	CALLS	#4, LIB\$STOP
	00000000G		00	95	00126	TSTB	FILE_NAM+59
			12	12	0012C	BNEQ	118
	01	00000000G	00	91	0012E	CMPB	FILE_NAM+60, #1
			09	12	00135	BNEQ	118
	01	00000000G	00	91	00137	CMPB	FILE_NAM+61, #1
			27	13	0013E	BEQL	128
F8	AD	00000000G	00	9B	00140	MOVZBW	FILE_FAB+52, DESC
FC	AD	00000000G	00	D0	00148	MOVL	FILE_FAB+44, DESC+4
	7E	0828	8F	3C	00150	MOVZWL	#2088, -(SP)
	F8		AD	9F	00155	PUSHAB	DESC
			01	DD	00158	PUSHL	#1
	007710FA		8F	DD	0015A	PUSHL	#7803130
00000000G	00		04	FB	00160	CALLS	#4, LIB\$STOP
17 00000000G	00		03	E1	00167	BBC	#3, FILE_NAM+53, 138
	00771352		8F	DD	0016F	PUSHL	#7803730
			56	DD	00175	PUSHL	R6
			01	DD	00177	PUSHL	#1
	0077134A		8F	DD	00179	PUSHL	#7803722
00000000G	00		04	FB	0017F	CALLS	#4, LIB\$STOP
	50	00000000G	00	9A	00186	MOVZBL	FILE_NAM+58, R0
	50	00000000G	00	C0	0018D	ADDL2	FILE_NAM+72, R0
	04	FF	A0	90	00194	MOVB	-1(R0), ENDCHAR
F8	AD	00000000G	00	9B	00199	MOVZBW	FILE_NAM+11, DESC
F8	AD		02	A2	001A1	SUBW2	#2, DESC
FC	AD	00000000G	00	D0	001A5	MOVL	FILE_NAM+12, DESC+4
	59	FC	AD	D0	001AD	MOVL	DESC+4, R9
	5A		59	D0	001B1	MOVL	R9, STR_PTR
	50	F8	AD	3C	001B4	MOVZWL	DESC, R0
	57		50	D0	001B8	MOVL	R0, STR_LEN
			58	D4	001BB	CLRL	PTR
	56	FF	A0	9E	001BD	MOVAB	-1(R0)[R9], EOS
6A	57 00000000'		03	39	001C2	MATCHC	#3, P.AAR, STR_LEN, (STR_PTR)
			03	13	001CB	BEQL	158
	53		03	D0	001CD	MOVL	#3, R3
			53	D7	001D0	DECL	R3
	5B		73	3E	001D2	MOVAB	-(R3), TEMP
			12	13	001D5	BEQL	168
	58		5B	D0	001D7	MOVL	TEMP, PTR
53	5A		5B	C3	001DA	SUBL3	TEMP, STR_PTR, R3
	57	FD	A3	47	9E	MOVAB	-3(R3)[STR_LEN], STR_LEN
	5A	03	AB	9E	001E3	MOVAB	3(R11), STR_PTR
			D9	11	001E7	BRB	148
			58	D5	001E9	TSTL	PTR
			03	13	001EB	BEQL	178
	58		03	C0	001ED	ADDL2	#3, PTR
6A	57		2E	3A	001F0	LOCC	#46, STR_LEN, (STR_PTR)
			02	12	001F4	BNEQ	188
			51	D4	001F6	CLRL	R1
	5B		51	D0	001F8	MOVL	R1, TEMP

				12	13	001FB	BEQL	198		
		58		58	D0	001FD	MOVL	TEMP, PTR		0758
53		5A		58	C3	00200	SUBL3	TEMP, STR_PTR, R3		0759
		57	FF A347	9E	00204	MOVAB	-1(R3)[STR_LEN], STR_LEN			
		5A	01	AB	9E	00209	MOVAB	1(R11), STR_PTR		0760
				E1	11	0020D	BRB	178		0751
				58	D5	0020F	TSTL	PTR		0763
				32	13	00211	BEQL	228		
		56		58	D1	00213	CMPL	PTR, EOS		0771
				07	12	00216	BNEQ	208		
				58	D6	00218	INCL	PTR		0774
		88		2A	90	0021A	MOVB	#42, (PTR)+		
				5F	11	0021D	BRB	248		0771
		5A		58	D1	0021F	CMPL	PTR, STR_PTR		0786
				14	12	00222	BNEQ	218		
01	57	56		58	C3	00224	SUBL3	PTR, EOS, STR_LEN		0789
	A8	68		57	28	00228	MOVC3	STR_LEN, (PTR), 1(PTR)		0790
		68	04	AE	90	0022D	MOVB	ENDCHAR, (PTR)		0791
		58	01	A748	9E	00231	MOVAB	1(STR_LEN)(PTR), PTR		0792
				46	11	00236	BRB	248		0786
	57	56		5A	C3	00238	SUBL3	STR_PTR, EOS, STR_LEN		0796
		88	04	AE	90	0023C	MOVB	ENDCHAR, (PTR)+		0797
		5A		57	C0	00240	ADDL2	STR_LEN, PTR		0798
				39	11	00243	BRB	248		0763
	6A	57		2C	3A	00245	LOCC	#44, STR_LEN, (STR_PTR)		0809
				02	12	00249	BNEQ	238		
				51	D4	0024B	CLRL	R1		
		6E		51	D0	0024D	MOVL	R1, STATUS		
		50	00000000G	00	9A	00250	MOVZBL	FILE NAM+57, R0		0813
		5A		01	A049	9E	MOVAB	1(R0)(R9), STR_PTR		
07	50	56		5A	C3	0025C	SUBL3	STR_PTR, EOS, R0		0814
	AA	6A		50	28	00260	MOVC3	R0, (STR_PTR), 7(STR_PTR)		
		58		53	D0	00265	MOVL	R3, TEMP		
	6A	EF	00000000'	06	28	00268	MOVC3	#6, P.AAS, (STR_PTR)		0815
		5A		53	D0	00270	MOVL	R3, STR_PTR		
		6A		04	AE	90	MOVB	ENDCHAR, (STR_PTR)		0816
				6E	D5	00277	TSTL	STATUS		0821
				05	12	00279	BNEQ	258		
		58		58	D0	0027B	MOVL	TEMP, PTR		
				70	11	0027E	BRB	288		
24	00	6E		00	2C	00280	MOVC5	#0, (SP), #0, #36, TPARSE_BLOCK		0830
				10	AE	00285				
				08	D0	00287	MOVL	#8, TPARSE_BLOCK		0831
18	AE	56	10	5A	C3	0028B	SUBL3	STR_PTR, EOS, TPARSE_BLOCK+8		0833
		53		AA	9E	00290	MOVAB	7(RT0), R3		0834
			1C	AE	53	D0	MOVL	R3, TPARSE_BLOCK+12		
				EF	9F	00298	PUSHAB	DIR_KEYS		0835
				EF	9F	0029E	PUSHAB	DIR_STATE		
			00000000'	AE	9F	002A4	PUSHAB	TPARSE_BLOCK		
			00000000'	03	FB	002A7	CALLS	#3, LIBSTPARSE		
		00000000G	00	50	D0	002AE	MOVL	R0, STATUS		
				6E	E9	002B1	BLBC	STATUS, 268		0842
			08	06	B0	002B4	MOVW	#6, TEMP_DESC		
			OC	53	D0	002B8	MOVL	R3, TEMP_DESC+4		0843
				00	DD	002BC	PUSHL	MEMBER		0848
				00	DD	002C2	PUSHL	GROUP		0847
				AE	9F	002C8	PUSHAB	TEMP_DESC		0844

SETDIR
V04-000

M 2
16-Sep-1984 00:59:34
14-Sep-1984 12:09:06

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETDIR.B32;1

Page 31
(8)

		14	AE 9F 002CB	PUSHAB	TEMP DESC	
		00000000'	EF 9F 002CE	PUSHAB	P.AAT	
00000000G	00		05 FB 002D4	CALLS	#5, SYSSFA0	
	6E		50 D0 002DB	MOVL	R0, STATUS	
	0B		6E E8 002DE	BLBS	STATUS, 27%	
			6E DD 002E1	PUSHL	STATUS	0849
00000000G	00		01 FB 002E3	CALLS	#1, LIB\$STOP	
			04 11 002EA	BRB	28%	
	58	0E	AA 9E 002EC	MOVAB	14(R10), PTR	0850
	88	00000000'	EF D0 002F0	MOVL	P.AAV, (PTR)+	0854
	58		59 83 002F7	SUBB3	R9, PTR, FILE_FAB+52	0855
00000000G 00			59 D0 002FF	MOVL	R9, FILE_FAB+44	0856
	00		01 D0 00306	MOVL	#1, R0	0858
	50		04 00309	RET		
			50 D4 0030A	CLRL	R0	0859
			04 0030C	RET		

; Routine Size: 781 bytes, Routine Base: \$CODE\$ + 03CF

SETDIR
V04-000

N 2
16-Sep-1984 00:59:34
14-Sep-1984 12:09:06

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SETDIR.B32;1

Page 32
(9)

: 868 0860 1 END
: 869 0861 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	100	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
-LIB\$KEYOS	0	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
-LIB\$STATES	16	NOVEC, NOWRT, RD, EXE, SHR, LCL, REL, CON, PIC, ALIGN(1)
\$CODES	1756	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$PLITS	208	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	109	0	1000	00:01.7
-\$255\$DUA28:[SYSLIB]TPAMAC.L32;1	42	18	42	14	00:00.1

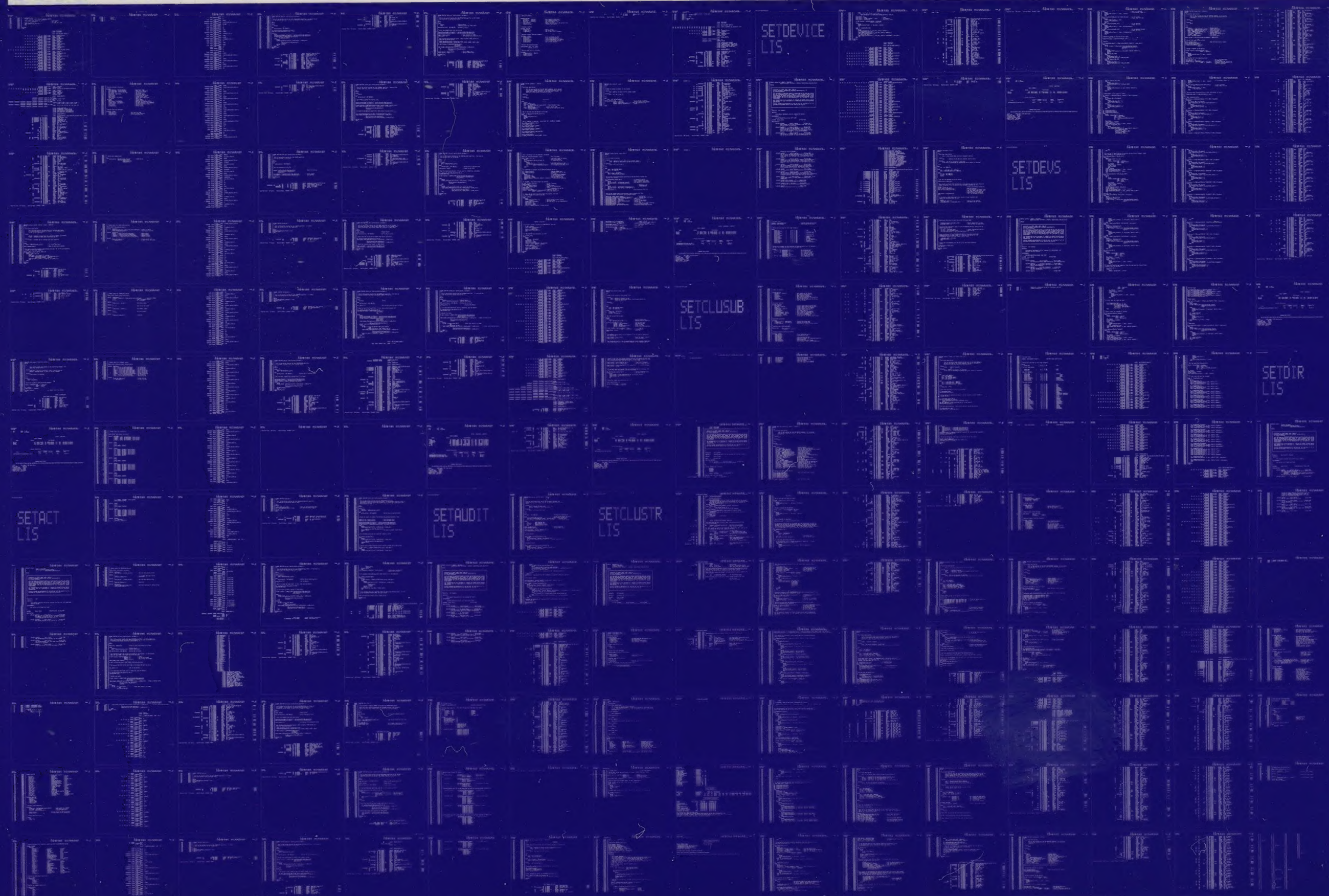
COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LISS:SETDIR/OBJ=OBJ\$:SETDIR MSRC\$:SETDIR/UPDATE=(ENHS:SETDIR)

: Size: 1756 code + 324 data bytes
: Run Time: 00:32.7
: Elapsed Time: 01:40.0
: Lines/CPU Min: 1581
: Lexemes/CPU-Min: 25423
: Memory Used: 274 pages
: Compilation Complete

0052 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0053 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

